What is Claimed:

1. A method of querying a database during database recovery, the method comprising:

reading a log file containing log file transactions to find incomplete transactions.

comparing the log file transactions with transactions reflected in the database to find unentered transactions;

inserting the unentered transactions into the database;

asserting a first lock type on a data item associated with each incomplete transaction;

removing a transaction reflected in the database if the transaction is not committed in the log file, wherein the removal of the transaction de-asserts a first lock type on a data item associated with the incomplete transaction; and

permitting a snapshot query of the database concurrent with the removal of an incomplete transaction, wherein if a first lock type is detected on a first data item, the snapshot query is delayed until the first lock type is de-asserted.

2. The method of claim 1, wherein the step of asserting a first lock type comprises asserting at least one of a row, a page, a table and an index..

3. The method of claim 1, wherein the step of asserting a first lock type comprises asserting a redo lock.

4. The method of claim 1, further comprising:

permitting a snapshot query of the database concurrent with the removal of an incomplete transaction, wherein if a second lock type is detected on a second transaction, a read from a previous version of the second transaction is performed.

5. The method of claim 4, wherein the second lock type comprises one of a read and a write lock.

6. A method of querying a database during database recovery using a snapshot of the database, the method comprising:

reading a log file containing log file transactions to find incomplete transactions;

comparing the log file transactions with transactions reflected in the database to find unentered transactions;

inserting the unentered transactions into the database;

removing an incomplete transaction from the database if the incomplete transaction is not committed in the log file;

permitting a snapshot query against the database concurrent with the removing of an incomplete transaction, wherein a concurrent but earlier started query asserted a lock on a data item and stored a first version of the data item; and

testing for a lock related to a data item of the snapshot query, wherein if a lock related to a data item corresponding to the snapshot query is detected, a read from a first version of the data item is performed.

7. The method of claim 6, wherein the step of permitting a snapshot query comprises asserting a one of a read and a write lock on the data item.

8. A method of querying a database during recovery of the database, the method comprising:

reading a log file containing log file transactions to find incomplete transactions;

identifying unentered transactions within the log file transactions;

inserting the unentered transactions into the database; wherein a corresponding version record is generated and a lock on a corresponding individual data item is asserted;

removing a transaction from the database if the transaction is not committed in the log file wherein the removal of the transaction de-asserts a lock on a corresponding individual data item;

permitting a snapshot query against the database concurrent with the removal of the incomplete transactions; and

testing for a commit time related to a data item corresponding to the new query; wherein if commit time is after the start time of the new query, a read from a previous version record of the data item is performed.

9. The method of claim 8, wherein inserting the unentered transactions into the database further comprises inserting the unentered transactions into a mirror database before a primary database fails.

10. The method of claim 9, wherein the unentered transactions are shipped from a primary database to a mirror database.

11. A system for querying a database during recovery of the database, the system comprising:

a processor having access to memory, the memory having instructions of a software component; and

a software component which, when executed:

reads a log file containing log file transactions to find incomplete transactions;

compares the log file transactions with transactions reflected in the database to find unentered transactions;

performs redo operations with the unentered transactions;

performs undo operations with uncommitted transactions;

permits a snapshot query against the database concurrent with the undo operations; and tests for a lock related to a data item of the snapshot query; wherein if a lock related to a redo operation is detected, a read from a previous version of the data item is performed.

12. The system of claim 11, wherein locks are asserted upon the redo operation of one of a row, a page, a table and an index.

13. The system of claim 11, further comprising a display on which to view query results.

14. A system for the querying of a data base during recovery of the database, the system comprising:

a mirror database where in the mirror database receives log file transactions from a primary database;

a database processor in the mirror database, the database processor having access to memory, the memory having instructions of software components; and

a software component which, when executed,

reads the log file transactions to identify incomplete transactions;

compares the log file transactions with transactions reflected in the database to find unentered transactions;

performs redo operations using the unentered transactions, the redo operation asserting locks, wherein a corresponding version record is generated and a lock on a corresponding individual data item is asserted;

performing undo operations if a transaction is not committed in the log file, wherein an undo operation de-asserts a lock on a corresponding individual data item;

permitting a snapshot query against the mirror database concurrent with undo operations; and testing for a lock related to a data item corresponding to the snapshot query; wherein if a lock is detected, a read from a previous version record of the data item is performed.

15. The system of claim 14, wherein performing redo operations further comprises performing redo operations on the mirror database before a primary database fails.

16. The system of claim 14, wherein performing undo operations occurs after a primary database fails.

17. A computer-readable medium having computer-executable instructions for performing a method to query a database during recovery of the database, the method comprising:

reading a log file containing log file transactions to find incomplete transactions.

comparing the log file transactions with transactions reflected in the database to find unentered transactions;

inserting the unentered transactions into the database;

asserting a first lock type on a data item associated with each incomplete transaction;

removing a transaction reflected in the database if the transaction is not committed in the log file, wherein the removal of the transaction de-asserts a first lock type on a data item associated with the incomplete transaction; and

permitting a snapshot query of the database concurrent with the removal of an incomplete transaction, wherein if a first lock type is detected on a first data item, the snapshot query is delayed until the first lock type is de-asserted.

18. The computer-readable medium of claim 17, the method further comprising:

permitting a snapshot query of the database concurrent with the removal of an incomplete transaction, wherein if a second lock type is detected on a second transaction, a read from a previous version of the second transaction is performed.

19. A computer-readable medium having computer-executable instructions for performing a method to query a database during recovery of the database using a snapshot of the database, the method comprising:

reading a log file containing log file transactions to find incomplete transactions;

comparing the log file transactions with transactions reflected in the database to find unentered transactions;

inserting the unentered transactions into the database;

removing an incomplete transaction from the database if the incomplete transaction is not committed in the log file;

permitting a snapshot query against the database concurrent with the removing of an incomplete transaction, wherein a concurrent but earlier started query asserted a lock on a data item and stored a first version of the data item; and

testing for a lock related to a data item of the snapshot query, wherein if a lock related to a data item corresponding to the snapshot query is detected, a read from a first version of the data item is performed.

20. The computer-readable medium of claim 19, wherein the step of permitting a snapshot query comprises asserting one of a read and a write lock on the data item.

21. A computer-readable medium having computer-executable instructions for performing a method to query a database during recovery of the database, the method comprising:

reading a log file containing log file transactions to find incomplete transactions;

identifying unentered transactions within the log file transactions;

inserting the unentered transactions into the database; wherein a corresponding version record is generated and a lock on a corresponding individual data item is asserted;

removing a transaction from the database if the transaction is not committed in the log file wherein the removal of the transaction de-asserts a lock on a corresponding individual data item;

permitting a snapshot query against the database concurrent with the removal of the incomplete transactions; and

testing for a commit time related to a data item corresponding to the new query; wherein if commit time is after the start time of the new query, a read from a previous version record of the data item is performed.

22. The computer-readable medium of claim 21, wherein the method step of inserting the unentered transactions into the database further comprises inserting the unentered transactions into a mirror database before a primary database fails.

23. The computer-readable medium of claim 21, wherein the unentered transactions are shipped from a primary database to a mirror database.

24. A system for querying a database during recovery of the database, the system comprising:

a processor having access to memory, the processor having means for performing a method comprising:

reading a log file containing log file transactions to find incomplete transactions;

comparing the log file transactions with transactions reflected in the database to find at least one unentered transaction;

conducting at least one redo operation with the at least one unentered transaction;

conducting at least one undo operation with at least one uncommitted transaction;

allowing a snapshot query against the database concurrent with the at least one undo operation; and testing for a lock related to a data item of the snapshot query; wherein if a lock related to a redo operation is detected, a read from a previous version of the data item is performed.

25. A system for the querying of a data base during recovery of the database, the system comprising:

a mirror database having means to receive log file transactions from a primary database;

a database processor in the mirror database, the database processor having means to:

read the log file transactions to identify incomplete transactions;

perform redo operations using unentered transactions, the redo operation asserting locks; wherein a corresponding version record is generated and a lock on a corresponding individual data item is asserted;

perform undo operations if a transaction is not committed in the log file, wherein an undo operation de-asserts a lock on a corresponding individual data item;

permit a snapshot query against the mirror database concurrent with undo operations; and

test for a lock related to a data item corresponding to the snapshot query; wherein if a lock is detected, a read from a previous version record of the data item is performed.